BMC Bioinformatics

RESEARCH

Open Access

Denoiseit: denoising gene expression data using rank based isolation trees

Jaemin Jeon[1], Youjeong Suk[2], Sang Cheol Kim[3], Hye-Yeong Jo[3], Kwangsoo Kim[4,5*] and Inuk Jung[2*]

*Correspondence:
kwangsookim@snu.ac.kr;
inukjung@knu.ac.kr

[1] Interdisciplinary Program in Bioinformatics, Seoul National University, Gwanak-gu, Seoul 08826, Republic of Korea
[2] School of Computer Science and Engineering, Kyungpook National University, Buk-gu, Daegu 41566, Republic of Korea
[3] Division of Healthcare and Artificial Intelligence, Department of Precision Medicine, Korea National Institute of Health, Korea Disease Control and Prevention Agency, Osong, CheongJu 28159, Republic of Korea
[4] Department of Transdisciplinary Medicine, Seoul National University Hospital, Jongno-gu, Seoul 03080, Republic of Korea
[5] Department of Medicine, Seoul National University, Jongno-gu, Seoul 03080, Republic of Korea

Abstract

Background: Selecting informative genes or eliminating uninformative ones before any downstream gene expression analysis is a standard task with great impact on the results. A carefully curated gene set significantly enhances the likelihood of identifying meaningful biomarkers.

Method: In contrast to the conventional forward gene search methods that focus on selecting highly informative genes, we propose a backward search method, DenoiseIt, that aims to remove potential outlier genes yielding a robust gene set with reduced noise. The gene set constructed by DenoiseIt is expected to capture biologically significant genes while pruning irrelevant ones to the greatest extent possible. Therefore, it also enhances the quality of downstream comparative gene expression analysis. DenoiseIt utilizes non-negative matrix factorization in conjunction with isolation forests to identify outlier rank features and remove their associated genes.

Results: DenoiseIt was applied to both bulk and single-cell RNA-seq data collected from TCGA and a COVID-19 cohort to show that it proficiently identified and removed genes exhibiting expression anomalies confined to specific samples rather than a known group. DenoiseIt also showed to reduce the level of technical noise while preserving a higher proportion of biologically relevant genes compared to existing methods. The DenoiseIt Software is publicly available on GitHub at https://github.com/cobi-git/DenoiseIt

Keywords: Gene, Noise, Filtering, Matrix factorization

Introduction

Gene expression analysis is a fundamental aspect of transcriptomic research, providing valuable insight into various biological processes and disease mechanisms. Especially, the differentially expressed genes (DEG) between two or more groups are of interest. It is a common practice to perform DEG analysis on a set of genes that are manually curated or collected by some gene selection method. In most cases, a gene selection method is used to compose a baseline gene set. The simple but effective Principal Component Analysis (PCA) is a prevalently used gene selection method where highly variably expressed genes are selected. However, the presence of noisy or outlier genes can significantly disrupt the integrity of such analysis, leading to inaccurate results. This

becomes more problematic when the number of samples is small since the gene expression variance of a population may not be well captured leading to poor gene selection decisions. As a gene set is input to many downstream analysis, it has non-trivial impact on their results.

Bulk and single-cell RNA sequencing has emerged as a pivotal tool in transcriptomic analysis and has seen significant improvements in accuracy. However, it is not devoid of challenges, particularly related to technical noise embedded in the data. Technical noise can arise from various factors such as RNA-seq library preparation, amplification, sequencing biases, or even random hexamer priming during the sequencing reaction. In the case of technical noise, a common approach for its mitigation involves removing genes that have predominantly low or zero values across the majority of samples. Biological noise is another source of variability in gene expression data arising from inherent stochasticity and complexity in biological systems. Unlike technical noise, which originates from experimental or technical factors, biological noise emerges from the natural fluctuations and variability in cellular processes. Biological noise can attribute to factors such as genetic regulatory mechanisms, cell-to-cell variability and environmental influences. In gene expression data, biological noise manifests as fluctuations in expression levels of genes even within a homogeneous population of cells. Researchers often grapple with both technical and biological noise when analyzing gene expression data. Statistical methods and computational techniques are employed to distinguish between these two sources of noise and to extract meaningful biological insight from the data. There is yet no standard procedure for removing both technical and biological noise and it remains to be improved.

There are a number of methods for removing noise from gene expression data prior to downstream analysis. Some examples are, applying log-transformation followed by techniques such as z-scoring or quantile normalization, removing anomalous sample data, and imputing missing values in single-cell RNA-seq data. While these strategies effectively mitigate noise, they often do not place significant emphasis on gene removal. Particularly in the case of sample removal, the importance of individual patient data in real-world applications makes it a challenging decision since valuable sample resources are not being utilized. Before taking such drastic steps of removing an entire sample, we can consider removing outlier genes first as it is possible to enhance the similarity between samples within the same group and thereby potentially preserving valuable samples and strengthening the overall power of statistical analysis. A substantial number of methods using statistical or machine learning methods were developed for the purpose of qualitative gene selection. The methods can be categorized into forward, backward and bi-directional search based on their direction of the gene selection process. In the forward search, the search starts with an empty set where informative genes are added to it [1]. In the backward search, uninformative genes are removed from the whole gene set until some criteria is met [2]. At last, the bi-directional search performs both forward and backward search in an iterative manner [3, 4]. The majority of gene selecting methods belong to the forward search category that includes the well known Principal Component Analysis (PCA) based gene selection method. While the gene set from forward search is conclusive, the backward search is used to remove unwanted noise on which forward

search can be applied for improving the quality of the baseline gene set. In this study, we focused on the backward search where uninformative genes are deemed as noise or uninformative.

Besides the directional search property, the gene selection methods can also be categorized into three types: wrapper, embedded, and filtering based. They differ from the directional gene selection methods in terms of being more machine learning based. Wrapper methods involve the use of models to evaluate the relevance of individual genes. These methods typically employ evaluation criteria such as cross-validation and prediction accuracy to assess the performance of selected gene subsets. Embedded methods, on the other hand, perform gene selection during the model training process. Evaluation criteria in embedded methods are directly related to the model's learning and generalization capabilities, including performance metrics and model complexity. Gene filtering stands as a distinct approach. In gene filtering, genes are selected based on predefined statistical or information-theoretic criteria, often involving methods like t-tests or ANOVA to determine the relevance of genes in relation to a specific condition or disease. For the gene filtering methods, various approaches exist for gene selection through unsupervised techniques or the removal of genes suspected to be outliers. Several methods have been developed to address noise in gene-level data generated during experiments. To reduce the impact of low-expressed genes and mitigate sample variance, techniques like noisyR [5], threshold-based gene removal [6], and MGSACO [7] eliminate these genes during data preprocessing. Another approach, OutSingle [8], employs Singular Value Decomposition (SVD) to calculate outlier scores for genes, identifying those with unusual behavior within the dataset. Principal Component Analysis (PCA) is used in the PCAUFE method [9] to compute p-values for gene selection. Additionally, gene clustering-based methods like kVirtuals [10] have been used for gene selection. However, current techniques primarily focus on eliminating low-expression genes and do not effectively identify and eliminate noisy genes specific to certain samples or patients. These noisy genes, often referred to as sample-biased genes, do not represent consistent expression patterns across the same sample group. The presence of such sample specific genes can hinder the accurate identification of biologically meaningful patterns and relationships, underscoring the importance of developing more sophisticated methods to discern and remove these potential sources of noise.

In this study, we propose a novel approach for removing such outlier genes to enhance the robustness of gene expression pattern analyses. Here, we propose a novel unsupervised backward search based gene removal method, DenoiseIt. The DenoiseIt method builds on the hypothesis that samples belonging to a common group, in terms of clinical or biological characteristics, should exhibit similar gene expression patterns. Here, we also assume that the group label of each sample are given, while they are only used for performance evaluation and not used during the gene removal process. To achieve this, we employ the non-negative matrix factorization (NMF) algorithm to first group patients with similar gene expression profile based on the rank features. Subsequently, genes that are specific to a single sample are removed from the candidate gene set using the isolation forest method. The isolation forest [11] method constructs a tree to identify such outliers genes. This procedure can be applied for multiple iterations. From the isolation trees, we can identify samples that deviate from their respective groups and

pinpoint the genes responsible for such discrepancy and thus remove them from the gene set.

To evaluate the performance of DenoiseIt, we utilized both bulk and single-cell RNA-seq data to assess its biological effectiveness. Furthermore, the gene removal performance of DenoiseIt was compared with three ranking based gene filtering methods (i.e., MGSACO, OutSingle, PCAUFE), two threshold based gene filtering methods (i.e., noisyR, kVirtuals) and the case where gene filtering was not performed. Furthermore, two wrapper based methods (i.e., SVM-RFE [12], SAFS [13]) were also included in our performance evaluation. As a result, we showed that DenoiseIt was able to improve the identification of cancer subtype specific genes in four cancer types. Also, it preserved more biologically meaningful genes specific to the level of severity when applied to a total of 456 single-cell RNA-seq samples from a COVID-19 cohort. It also showed robust results when the number of samples between the groups were unbalanced.

Materials and methods

Data

Our analysis incorporates two datasets: The Cancer Genome Atlas (TCGA) public dataset [14] and the COVID-19 dataset [15]. The TCGA public dataset is a well-established and widely used resource for cancer genomics research. Here, we utilized the gene expression profiles of four cancer types, which are COAD (Colon adenocarcinoma, n = 222), STAD (Stomach adenocarcinoma, n = 305), BRCA (Breast invasive carcinoma, n = 595) and LUAD (Lung adenocarcinoma, n = 180) that were initially comprised of 56,000 genes. Genes with less than an average expression count of five were excluded, resulting in 24852, 24913, 25428 and 24913 genes in the COAD, STAD, BRCA and LUAD datasets, respectively.

The COVID-19 dataset is a multi-omics dataset that includes gene expression profiles of peripheral blood mononuclear cells (PBMCs) from COVID-19 infected patients in South Korea. The COVID-19 dataset encompasses single-cell RNA-seq data from 456 samples comprised of 20212, 20875, 20541, 19541 and 16874 genes in each of the CD4 T, CD8 T, Monocyte, Natural Killer (NK) and B cell type, respectively. To objectively categorize the severity of each sample, the World Health Organization score (WHO [16]) was utilized as our metric. The maximum WHO score attained by each sample during their hospital stay served as the benchmark for labeling severity. Specifically, samples with a maximum WHO score exceeding 5 were classified as "Severe"(n = 92), while those with a maximum score of 5 or below were designated as "Moderate"(n = 364). This stratification methodology was implemented to ensure a uniform and fair assessment of sample severity across the cohort. The COVID-19 dataset was preprocessed by first annotating the cell type of each scRNA-seq sample using Azimuth [17], then aggregating each cell type's single-cell RNA-seq sample into a single pseudobulk sample. At the cell type level 1 of the Azimuth reference dataset, six cell types are present: CD4 T, CD8 T, NK, B, Monocyte and dendritic cells. Among those, the dendritic cell type exhibited less than three cell counts in all the samples, and thus was not considered for further analysis. A total of 1,364,590 cells remained, that were distributed as follows: CD4 T = 160,646, CD8 T = 174,239, B = 145,866, NK = 157,844 and Monocyte = 599,344. Finally, each cell type specific pseudobulk sample was subject to trimmed mean of

M-values (TMM) normalization. A pseudobulk sample was made by the aggregated sum of the gene expression counts for each cell type. For the further evaluation, simulated gene expression count data were generated. We created samples that follow the negative binomial distribution for 20,000 genes varying sample sizes of n = 25, 50, 75, 100, 150, 200, 300, and 400. All the simulated samples were structured into two groups to reflect two distinct conditions where genes were differentiated based on their ability to distinguish between these groups, characterized by a true log fold change (logFC) value. The average expression count of genes was set to 4 with a variance of 5. In addition, simulation data with no variance (i.e., no noise) between the samples within each group were generated to observe how many genes are retained by the various gene removal methods. For such purpose, the variance was set to 1, and the average expression count was set to 4.

Performance evaluation

Technical evaluation was performed on the noise removed gene set output from DenoiseIt by comparing it with various unsupervised gene filtering methods, noisyR, PCAUFE, OutSingle, MGSACO and kVirtuals. For both TCGA and COVID-19 datasets, here the evaluation criteria was how well the noise removed gene set of each method was able to discriminate the cancer subtypes or the severity of COVID-19 patients per cell type, respectively. Below, a brief description of three gene selection methods are described that are used for the performance evaluation. The methods were further compared for biological correctness via DEG and pathway analysis. DEG analysis was performed on each method's output gene set to investigate how well the DEGs from each gene set captured dataset related pathways. DESeq2 and edgeR were used for the DEG analysis.

Threshold based gene filtering method

The noisyR is a backward search method that reduces the impact of low-abundance genes on differential expression analysis. It uses read count values to determine the similarity between samples and applies a threshold to filter out genes that contribute to noise. kVirtuals is a forward search method selecting only small number of genes based on gene clustering using Normalized Mutual Information (NMI). Both method had their own threshold for filtering genes. PCAUFE is a PCA based unsupervised gene selection method. From the PCAUFE results, PC1 and PC2 were used to select genes with an adjusted p-value below 0.05.

Gene ranking based filtering method

OutSingle provides gene selections based on p-value, while MGSACO outputs genes by specifying the desired number of genes to be retained. To ensure a fair comparison, we selected the same number of genes as DenoiseIt for each dataset.

Wrapper method

Wrapper methods iteratively add or remove features to optimize the model's performance. Similarly, SAFS performs iterative clustering to enhance the clustering quality by adding or removing genes from the dataset. Unlike other methods, SVM-RFE is a

supervised method, thus it is not specifically applicable for gene selection or filtering when sample labels are unknown. Nevertheless, it was included to observe its performance in comparison to the other unsupervised methods.

No gene filtering

To evaluate the methods performance compared to the case without any gene filtering, we employed the full set of genes available in the datasets. Genes with expression values of zero across all samples were removed. Subsequently, the remaining data underwent log_2 transformation, normalization and then min-max scaling to ensure consistency and to mitigate potential biases in the data. The TCGA expression data were quantile normalized, whereas TMM was used for single-cell data.

Workflow of DenoiseIt

DenoiseIt is a novel approach for removing outlier genes from gene expression data. The rationale behind DenoiseIt is grounded in the hypothesis that samples within a same group should display similar expression patterns. Here, a group refers to a set of patients or samples with similar phenotypic background. Using NMF [18], we can easily observe whether the sample groups are well captured by the rank features. More importantly, by observing the rank features we can identify samples that deviate from their respective groups, along with the genes responsible for such discrepancies.

DenoiseIt is comprised of three stages: (1) NMF analysis, (2) outlier score computation and (3) outlier detection and removal (Fig. 1). In the first stage, the gene expression data is subject to NMF. The output of NMF are two matrices which serve as the input for the subsequent steps. In the second stage, outlier scores are computed using the loading and basis matrices from the NMF output. Here, isolation forests are generated on the decomposed ranks in H to compute the outlier scores, which effectively quantifies the likelihood of a rank being an outlier. In the last stage, genes associated to outlier ranks are identified using W and removed. This approach can also be applied for identifying outlier rank associated samples and prune any outlier samples instead of genes.

Stage 1: NMF analysis

First, genes with an average read count value of less than 5 were removed. The expression levels of the remaining genes were then log_2 transformed and quantile normalized. Consider the gene expression dataset as a $n \times m$ matrix K, where n and m refer to the number of genes and samples in matrix K respectively. Each column in K corresponds to a sample s_j for $j = 1, 2, ..., m$, and each row corresponds to a gene g_i for $i = 1, 2, ..., n$. After performing NMF on matrix K, two non-negative matrices W ($n \times q$ matrix) and H ($q \times m$ matrix) are obtained, where q is the number of ranks, which are denoted as r_t for $t = 1, 2, ...q$, such that $K \approx WH$. Here, W and H represent the basis (gene component) and the coefficient (sample component) matrices, respectively.

Stage 2: Outlier score computation

For each rank feature t, we utilize the Isolation Forest algorithm [11] to calculate an outlier score for each sample. Isolation Forest is an ensemble method that generates binary trees to isolate instances considered as outliers. The core idea behind the Isolation Forest

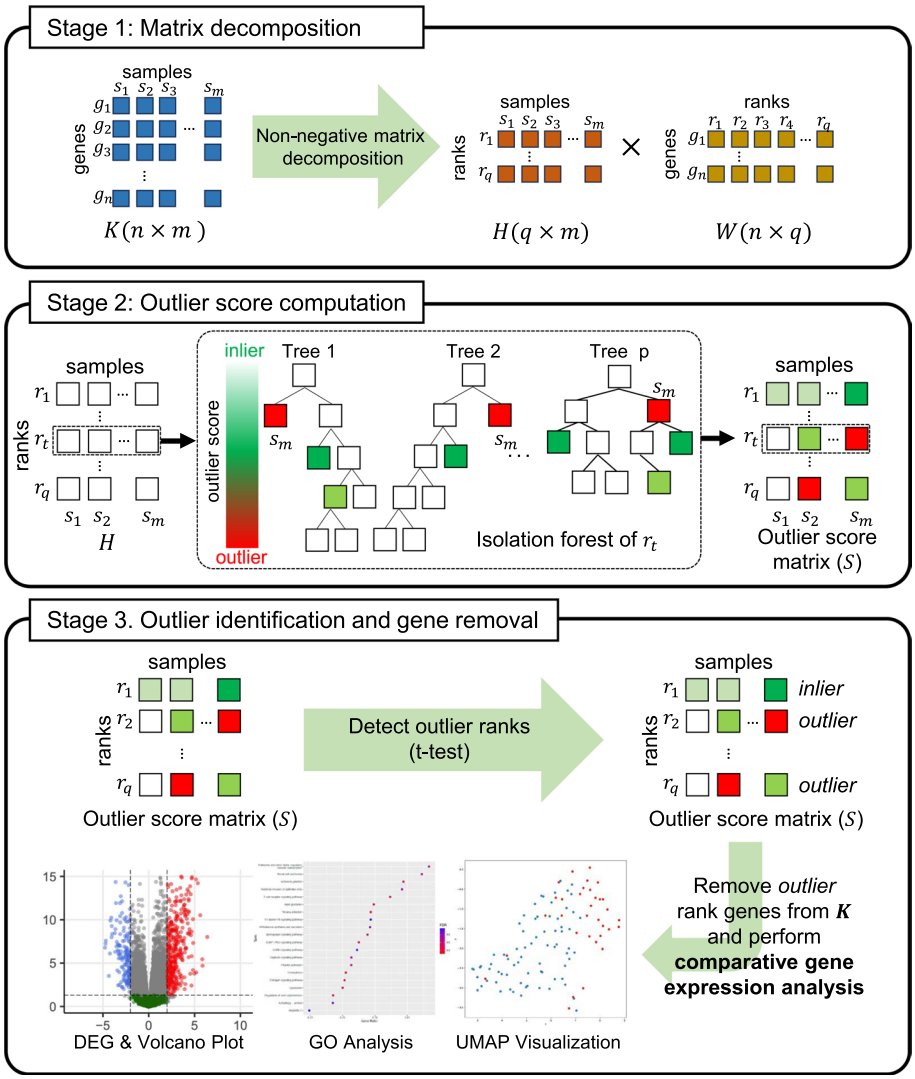

Fig. 1 DenoiseIt consists of three stages. First, it processes the gene expression data and decomposes it into basis and loading matrices using NMF. In the second step, each rank feature from the decomposed result are used to generate isolation trees to compute its outlier score. Finally, ranks are labeled as either inliers and outliers where genes associated to outlier ranks are removed. The remaining genes are used as input to various downstream analysis

is that anomalous data points are easier to isolate than inlier data points. The algorithm randomly selects a feature and randomly sets a split value between the maximum and minimum values of the selected feature. This process is repeated until the data points are isolated or the maximum number of tree depths is reached. The algorithm then computes an outlier score for each instance based on the average path length from the root node to the outlier node in all trees.

The output of the Isolation Forest algorithm is the outlier score matrix $S = q \times m$. Here, an isolation tree T_l is constructed for each rank t using H. A total of p number of trees, $l = 1, 2, ..., p$, are constructed for a single rank t to compute an outlier score for each sample j, which is denoted as $S_{t,j}$. The computation of an outlier score of $S_{r,s}$ can be encapsulated as follows:

Let's assume a randomly selected subset of samples s' is used to construct a tree T_l for rank t. The nodes of T_l are samples. Starting from the root, a split point x is randomly selected between the range of $min(H_t)$ and $max(H_t)$ to split the samples in tree T_l. The selection of split point x is done for each split w. The splitting continues until each sample is a leaf node or the maximum tree depth d is reached. The maximum tree depth is set to $d = \lceil \log_2(|s'|) \rceil$ to ensure that all samples can be isolated if they were the only ones in their leaves.

$$SPLIT_w = \begin{cases} s_j \in s' : H_{t,j} < x_w, \text{ if } s_j \text{ in left child node} \\ s_j \in s' : H_{t,j} \geq x_w, \text{ if } s_j \text{ in right child node} \end{cases} \tag{1}$$

If a sample j is frequently located near to the root while other samples are evenly distributed within the tree in all trees, then the distance from sample j to the root against the average distance of other samples to the root will be statistically significant. Such property implies that samples such as j have very different rank values than the other samples and thus is a candidate for being an outlier. The distance of sample j to the root in tree l is defined as $L_{s_j}^{(T_l)}$. Then, the average distance of s_j to the root in all trees is $A_{t,j}$ and defined as follows:

$$A_{t,j} = \frac{1}{p} \sum_{l=1}^{p} L_{s_j}^{(T_l)} \tag{2}$$

The average distance is normalized in respect to the maximum distance from the root to a leaf node, which gives us the final outlier score of a sample j in rank t, $S_{t,j}$, as below.

$$S_{t,j} = \frac{A_{t,j} - \min(A)}{\max(A) - \min(A)} \tag{3}$$

Here, $\min(A)$ and $\max(A)$ represent the minimum and maximum anomaly scores over all the samples, trees and ranks. Since we aim to identify outlier ranks, this normalization ensures that the outlier scores are comparable across the ranks.

Stage 3: Outlier identification and gene removal

Once the outlier scores have been calculated, the next step is to identify and remove outlier ranks. Here, the one sample t-test is used to check whether the average outlier score of a sample j for a given rank t significantly deviates from the average outlier score across all rank features in S. For a rank t, the hypothesis and null hypothesis are

$$H_0 : \mu_{S_t} = \mu_{S_{all}} \tag{4}$$

$$H_a : \mu_{S_t} \neq \mu_{S_{all}} \tag{5}$$

where μ_{S_t} is the average outlier score for the rank t, and $\mu_{S_{all}}$ is the average outlier score across all ranks. If the p-value is significant (i.e,. < 0.05), we reject the null hypothesis and consider the rank t as an outlier. Once the outlier ranks are identified, we proceed with the gene removal process. Here, each gene in the W is assigned to a single rank with maximum rank value. The primary gene candidates for removal are those that are

assigned to the outlier ranks. Given that the genes exhibit a unique expression pattern only within a particular set of samples, their removal could help improve the overall robustness and reliability of the analysis by reducing the gene expression variance within a sample group.

Results

Technical performance evaluation

Our primary objective was to demonstrate the effectiveness of DenoiseIt in achieving more accurate and biologically meaningful clustering results compared to other existing models.

In Fig. 2 and Supplementary Figure S1, we compared clustering performance of each method using Adjusted Rand Index (ARI) [19] and silhouette score [20]. Here, the task was to cluster cells of the same cell type and cancer samples of the same subtype. The ARI is calculated between the predicted labels obtained by performing K-means clustering and the actual labels of each sample in the dataset, which are the molecular subtypes in the cancer datasets and the severity groups in the COVID-19 dataset. Given N number of samples and multiple number of sample clusters, the silhouette coefficient is calculated as follows: for each sample i, let $c1(i)$ be the average distance between

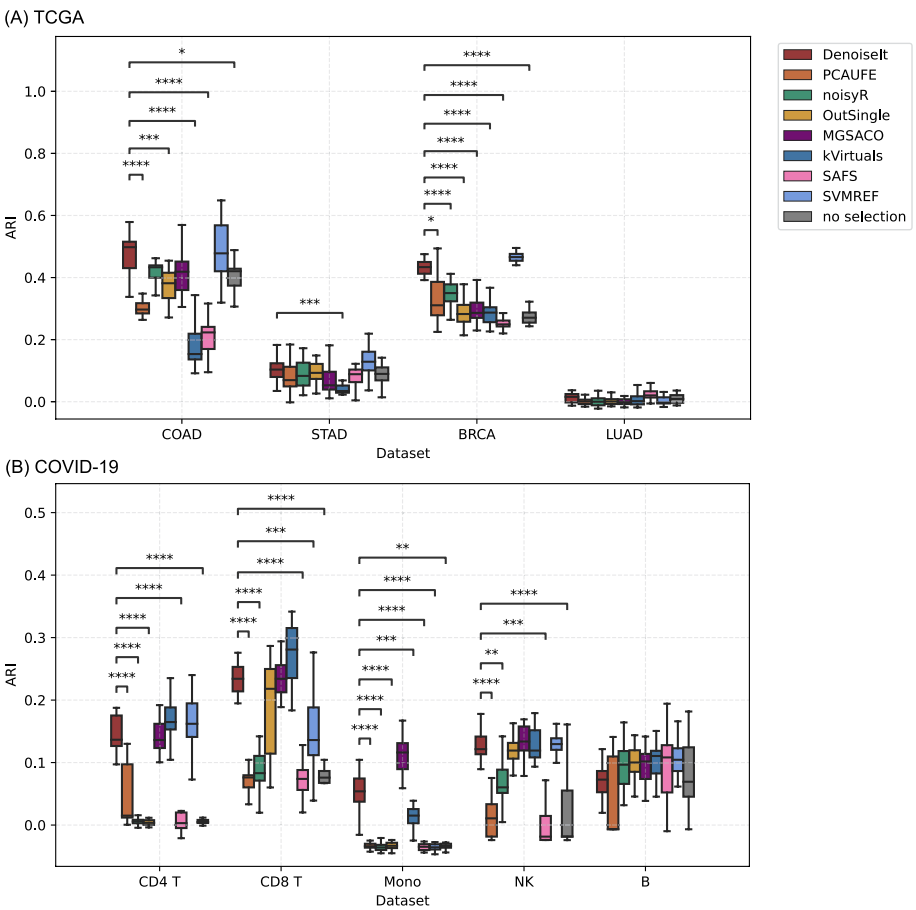

Fig. 2 The comparison of the Adjusted Rand Index (ARI) of **A** the TCGA, **B** COVID-19 dataset achieved through 20 iterations of random sampling, coupled with K-means clustering, initialized three times

sample i and all other samples in the same cluster, and let $c2(i)$ be the average distance between sample i and the samples in the nearest neighboring cluster. Once the silhouette coefficient is computed for all the samples, the mean of these coefficients provides the silhouette coefficient representing each dataset as shown in Eq. 6. Additionally, using the COVID-19 dataset, we balanced the number of samples of the severity groups and assessed the performance all the gene selection methods by randomly selecting 20 healthy and 20 severe patients. The result of the performance comparison between the tools are provided in Supplementary Figure S2.

$$meansilhouettescore = \frac{\sum_{N}^{i=1} \frac{c1(i)-c2(i)}{max(c1(i),c2(i))}}{N} \tag{6}$$

Cell type annotation was conducted using a reference single-cell dataset of PBMCs, and cells with a prediction score exceeding 0.6 were retained through the Seurat [21]. The classical K-means clustering was used. To preserve robustness, we randomly sampled 80% of the samples and computed the ARI for 20 times. Our results consistently showed that DenoiseIt outperforms other methods across all datasets in terms of ARI, emphasizing its superior and robust capability in identifying biologically meaningful clusters. These results further corroborate the superiority of DenoiseIt over other gene filtering methods in achieving more accurate clustering of samples. To ensure that the filtered genes do not negatively impact the performance of predicting subtypes, we conducted a 10-fold cross-validation test on both datasets. These tests were aimed at predicting cancer subtypes in TCGA and severity levels in COVID-19 datasets, as detailed in Supplementary Tables S1, S2. For these predictions, we employed logistic regression and random forest classifiers for each dataset. Using the expression data of the remaining genes, we predicted the subtype and severity labels for each sample. It was observed that all of the methods demonstrated similar performance, with no significant variation in values.

Biological performance evaluation

Here, we aimed to evaluate the effectiveness of DenoiseIt in retaining biologically meaningful genes, particularly DEGs, after the gene filtering process, compared to other methods. To do this, we first conducted a differential expression analysis using the entire genes in the dataset. DEG analysis was carried out using both the DESeq2 and edgeR methods for each subtype within the dataset, comparing them in a pairwise manner. Genes were identified as DEGs if they had an absolute log fold change (|logFC|) value greater than 1 and a False Discovery Rate (FDR) value of less than 0.05.

The key aspect of our evaluation was to determine how well each gene filtering method retains these identified DEGs. This is crucial because while gene filtering is essential for reducing noise and enhancing computational efficiency, preserving DEGs is important due to their likely substantial biological relevance to the phenotypes under study. Thus, the percentage of genes shared between the noise removed gene sets and the DEGs identified using the entire gene set was measured. Figure 3 and Supplementary Figure S3 illustrates the average percentage of overlapping genes between the DEGs and the genes retained by each method across various datasets. The percentage is calculated as the number of DEGs retained divided by the total number of genes remaining

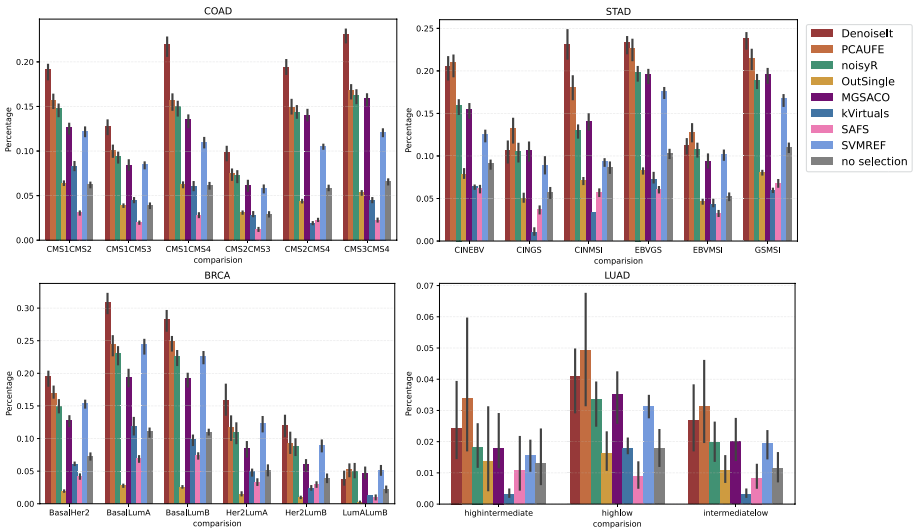

Fig. 3 The percentage of DEGs retained by the output of each gene filtering method per dataset that intersect with the DEGs identified without any gene filtering. While the performance of SVM-RFE is high, it must be noted that it is the only supervised based gene filtering method, which should not be directly compared to the other unsupervised methods

Table 1 Gene filtering results across all genes and calculated the proportion of genes retained after this process

Sample size	$n = 25$	$n = 50$	$n = 75$	$n = 100$	$n = 150$	$n = 200$	$n = 300$	$n = 400$
DenoiseIt	**0.9738**	**0.8835**	**0.8843**	**0.9144**	**0.8927**	0.8461	**0.8561**	**0.7843**
PCAUFE	0.8351	0.8392	0.8381	0.8957	0.8779	**0.8524**	0.8427	0.7392
noisyR	0.3587	0.4062	0.4324	0.4144	0.5321	0.5682	0.5078	0.5794
Outsingle	0.9381	0.8451	0.8432	0.8909	0.8810	0.8446	0.8491	0.7412
MGSACO	0.9372	0.8731	0.8551	0.8883	0.7931	0.7823	0.8358	0.7339

The method with best performance, or portion of retained DEG candidates, is highlighted for each n

after filtering (Retained DEGs/Retained Genes). This metric provides a direct measure of each method's ability to maintain biologically significant genes. Our results show that DenoiseIt, in particular, effectively retains a significant proportion of DEGs, indicating its efficiency not only in reducing dataset dimension but also in preserving crucial biological information.

We further compared the performance between the methods in terms of how many essential, or unnoisy, genes were retained across samples in the simulated bulk gene expression datasets with varying sample sizes of n = 25, 50, 75, 100, 150, 200, 300, and 400. For this assessment, we performed the evaluation under the assumption that genes demonstrating an absolute real log fold change greater than 1 between conditions, and those with variation less than or equal to 3, are DEG candidates and thus should not be eliminated from the dataset. Consequently, the efficacy of each method was compared based on the overlap between the genes retained by each method and the presumed candidate DEGs, as illustrated in Table 1.

In case if no noise exists in a dataset, it is natural to test each gene for significant gene expression difference and thus a gene removal method should retain all genes.

So, additionally, simulation gene expression data devoid of noise were generated, to ascertain the extent to which each method succeeds in preserving the most number of genes. The result of this evaluation are presented in Table 2. We only used gene filtering methods to ensure they do not filter useful genes. Since the Outsingle method does not provide a threshold for gene filtering it was not included in this evaluation.

$$\text{P-score} = \frac{1}{F}\sum_{i=1}^{F}\frac{1}{-\log_2\left(\frac{\text{overlap genes}_i}{\text{len(DEG)}}\right)} \tag{7}$$

To further demonstrate the effectiveness of each gene filtering method, including DenoiseIt, we performed downstream analyses using the gene sets that were output from each tool. One of the most common downstream analyses in transcriptomics is pathway analysis, which involves identifying enriched biological pathways from the list of DEGs. In this context, it is critical that the gene filtering method retains not just a large number of DEGs, but specifically those DEGs that are members of biologically meaningful pathways. The kVirtuals method is constrained to output 1000 or less genes, and thus was not included in this evaluation. Pathway enrichment analysis was performed using the Reactome [22] and Gene Ontology databases [23]. Also we used ImmuneSigDB [24] for COVID-19 dataset. The overlap between the list of DEGs and the genes in each enriched pathway was quantified using the Fisher's exact test. The percentage provides a measure of how much the set of retained genes overlaps with the DEGs in each enriched pathway. To assess the efficacy with which the remaining genes represent the pathway, we introduced a new score [5], which we inverted for better visualization. This 'P-score' compares the overlap of genes with a p-value less than 0.05 and an absolute log fold change (logFC) greater than 1, derived from the remaining genes, against the genes in the pathways using a hypergeometric test. This comparison aids in understanding the involvement of overlapping genes in the pathways. The P-score is calculated using the following equation, where F is the number of pathways with a p-value under 0.05. The comparison of the P-scores among different methods for the COAD, STAD and BRCA datasets is shown in Fig. 4. For the COVID-19 dataset, DenoiseIt showed robust performance in every cell type (Supplementary Figure S4). As shown, DenoiseIt consistently demonstrates robust and better performance in terms of preserving genes that are biologically relevant to the phenotypes of interest, thereby leading to more meaningful and interpretable results in downstream pathway analysis especially with ImmuneSigDB. This further showed that DenoiseIt

Table 2 Percentage of remained genes simulated datasets with low variance

Sample size	$n = 25$	$n = 50$	$n = 75$	$n = 100$	$n = 150$	$n = 200$	$n = 300$	$n = 400$
DenoiseIt	**1.0**	**1.0**	**0.9948**	**1.0**	**0.9959**	**0.9785**	**0.9608**	**0.9998**
PCAUFE	0.9732	0.9351	0.9151	0.9001	0.9204	0.9005	0.8892	0.9532
noisyR	0.3023	0.3047	0.3069	0.3095	0.4575	0.3532	0.3612	0.4389
MGSACO	0.7539	0.7351	0.7532	0.8301	0.7533	0.7433	0.7535	0.7884

The method with best performance, or portion of retained DEG candidates, is highlighted for each n

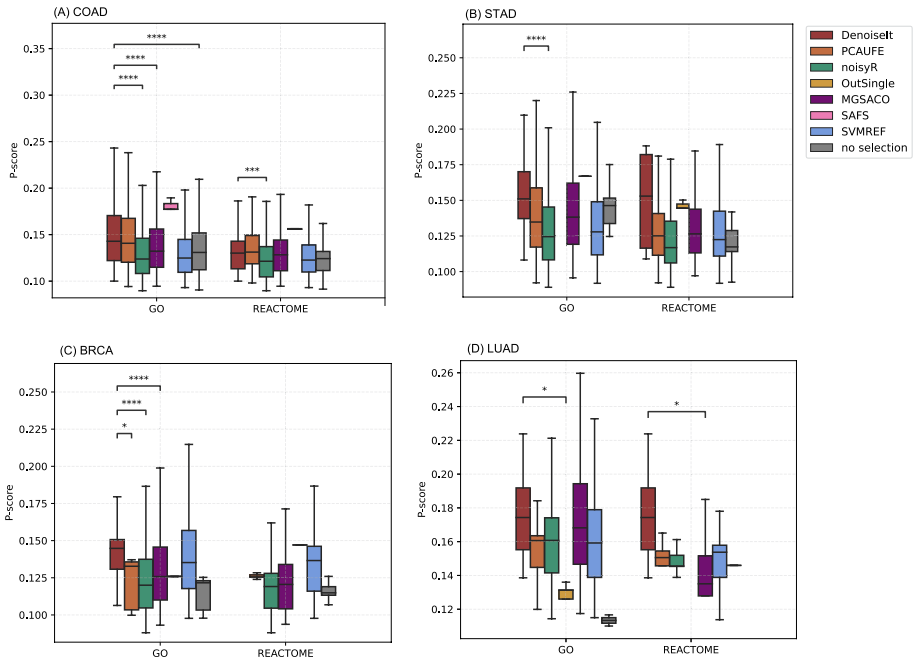

Fig. 4 The P-score of GO and pathway analysis using the refined gene sets of the methods using the TCGA dataset

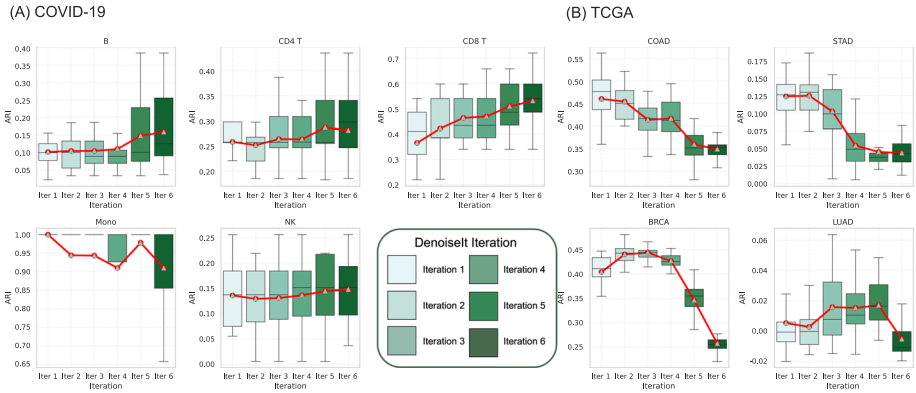

Fig. 5 Representation of the results of applying DenoiseIt for gene filtering across multiple iterations and the subsequent calculation of the ARI using **A** the COVID-19 and **B** TCGA dataset

provides an effective balance between noise reduction and biological signal preservation. PCAUFE showed better performance in the COAD dataset.

To investigate on the iterative adaptation of DenoiseIt, we conducted an exhaustive evaluation across multiple iterations. At the end of each iteration, we applied quantile normalization to the remaining genes, followed by the same gene-filtering process used in the previous steps. A total of six iterative gene filtering was performed for each data. The ARI was calculated after each iteration to measure the quality of clustering and thereby assess the robustness of the gene removal. The ARI of each iteration for each of dataset is shown in Fig. 5, where the remaining number of genes after filtering are shown in Table 3. We observed a general increase in ARI values,

Table 3 Number of remaining genes per iteration with DenoiseIt

Dataset	Type	gene count (iter=1)	gene count (iter=2)	gene count (iter=3)	gene count (iter=4)	gene count (iter=5)	gene count (iter=6)
COVID-19	B	6332	4337	2835	2184	1712	1308
	CD4 T	8710	7000	5343	2984	1485	651
	CD8 T	6836	4964	3372	2883	2232	1747
	Monocyte	8057	5288	3705	2564	1835	1355
	NK	6149	4801	3795	3047	2617	2072
TCGA	COAD	13,329	8710	5343	2984	1485	651
	STAD	13,500	7517	3952	2099	379	379
	BRCA	7867	6710	5343	2984	197	84
	LUAD	1,1026	6828	4109	2569	1187	484

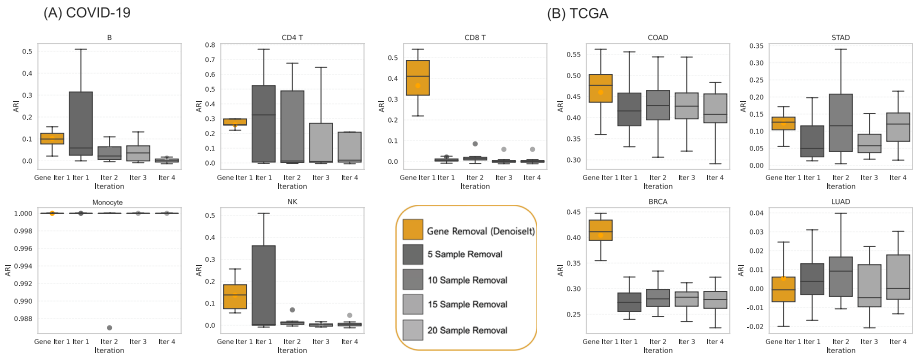

Fig. 6 The comparison between the effects of sample filtering and gene filtering utilizing the Adjusted Rand Index (ARI) in **A** the COVID-19 and **B** TCGA dataset

predominantly in the first 2-3 iterations, which suggests that the method becomes more robust with incremental refinement. However, the trend showed to be data specific in a few cases. By our experience, the results were generally satisfactory when the gene removal process was repeated up to three iterations. For instance, in the case of the COVID-19 single cell RNA-seq dataset, a continuous increase in ARI was observed even beyond the 3rd iteration. This was in contrast to the TCGA data, where the ARI tended to stabilize sooner. This discrepancy could be attributed to the higher noise levels in the single-cell data, where the iterative nature of DenoiseIt was particularly beneficial in progressively filtering out noisy genes. To dissect the role of gene filtering versus sample removal, we executed four iterations, removing 5 samples in each iteration based on their outlier ranks as determined by DenoiseIt. For this analysis, all genes were retained to isolate the effect of sample removal. Meanwhile, the impact of gene filtering was observed with the usual iterative process of DenoiseIt, where genes were selectively retained or removed. The samples in the case of sample removal were chosen based on their outlier rank in DenoiseIt, ensuring that we remove those with the highest outlier scores as shown in Fig. 6.

We found that while both gene filtering and sample removal have impacts on the final clustering and downstream analyses, their impacts vary in magnitude and

quality. The effect of sample removal was more pronounced in terms of the clustering metrics, suggesting that removing outlier samples can have immediate consequences on the clustering results. Collectively, gene filtering by DenoiseIt provided more improvement in capturing biologically relevant information, especially over multiple iterations. In the case of normalization, we observed that the iterative gene filtering process inherently led to more stable normalization statistics across the iterations. This suggests that by reducing the dimension and variability of the data, DenoiseIt indirectly aids in effective normalization.

In addition to comparing the performance of different gene selection tools, we also conducted a time comparison to evaluate their efficiency. For this comparison, we used a dataset consisting of 400 samples and restricted the computation to a single CPU to ensure a standardized testing environment (Supplementary Figure S5). The time taken by each tool to complete its process was measured in seconds. This approach allowed us to assess not only the effectiveness of the tools in selecting genes but also their computational efficiency, which is crucial for practical applications, especially when dealing with large datasets. Among the tools evaluated, PCAUFE and DenoiseIt demonstrated notably faster performance.

Discussion

In this study, we propose DenoiseIt, a novel approach for removing outlier genes leveraging NMF and the Isolation Forest algorithm, which outperformed competing methods in identifying and eliminating outlier genes, enhancing the reliability and interpretation of gene expression pattern analyses. Its performance was not only validated through comparative analyses, as depicted by ARI, but are also substantiated through its adept capability in preserving biologically significant genes that are intrinsic to specific cell types and conditions.

DenoiseIt offers several key advantages, including the ability to systematically identify outlier genes, enhance the robustness of gene expression pattern analyses, and mitigate the risk of inaccuracies in downstream analysis. Moving forward, our method holds the potential to be extended to outlier sample detection and removal, further enhancing the cleansing of gene expression datasets. By fostering more accurate and reliable analyses, our approach contributes to the advancement of biomedical research and our understanding of complex biological systems. DenoiseIt showcased its meticulousness in the context of the COVID-19 dataset, where it maintained marker genes corresponding to specific cell types, a feature vital for understanding the biological underpinnings and variances among different cells. This percentage in retaining cell-type specific marker genes is crucial as it aids in delineating the heterogeneous cellular landscape and comprehending the nuanced interplay of cells in disease conditions, like COVID-19. The approachâ€™s adaptability in refining gene selection iteratively, especially in datasets with varying noise levels and characteristics, underscores its versatile applicability in diverse research settings. In the case of BRCA, DenoiseIt uniquely retained around 1566 genes that were not preserved by other methods like noisyR or other methods (Fig. 7). Intriguingly, a substantial number of these uniquely retained genes were found to be closely associated with BRCA, and thus providing more room for investigating profound insights into the molecular mechanisms and pathways involved in breast cancer.

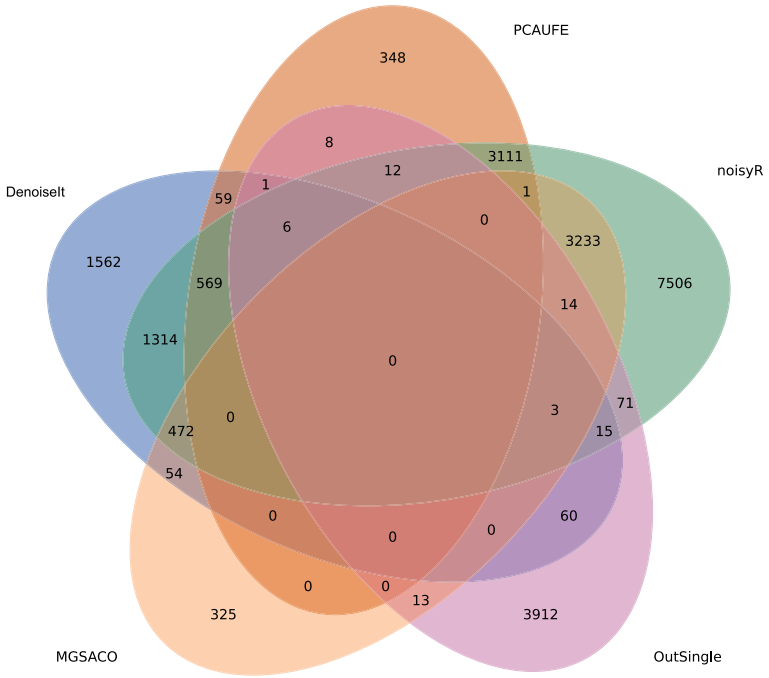

Fig. 7 The Venn diagram comparing the remaining genes identified by four different methods. This illustration provides a visual representation of the unique and shared genes retained by each method, highlighting the distinctiveness and overlap between them

More specifically, genes such as YWHAZ, LDHC, SYPL1P2, ABCB4, ZSCAN1, and others emerged as notable genes. For instance, YWHAZ showed a significant increase in BRCA and appeared to be less crucial in other cancers [25]. Silencing of LDHC in breast cancer cell lines significantly highlighted the presence of giant cells [26], demonstrating its importance in understanding the cellular dynamics in breast cancer. SYPL1P2 expressed universally across all analyzed cancer libraries but remained absent in normal tissues, suggesting its role as a potential marker for malignancy [27]. ABCB4's overexpression was related to acquired doxorubicin resistance in breast cancer cells in vitro [28], and ZSCAN1 played a role as a novel stemness-related tumor suppressor, targeting TAZ in breast cancer [29]. The presence of these genes, particularly when analyzed using DenoiseIt, provides a more in-depth understanding of the molecular architecture and genetic variations in BRCA, offering new perspectives and avenues for research in unraveling the complex nature of breast cancer and possibly other related cancers. We compared the remaining genes of DenoiseIt with those retained by other methods by evaluating their clustering performance which results are provided in Supplementary Table S3. In the case of the COVID-19 dataset, the number of genes retained by each method was relatively small. Therefore, we compared the percentage of overlapping genes with the DEGs identified before gene filtering (adjusted p-value < 0.05, $|\text{logFC}| > 1$). The preservation of such significant genes exemplifies DenoiseItâ€™s capability in not just reducing noise and dimensionality but also in extracting and preserving biologically relevant information, enhancing the depth and breadth of the subsequent analyses.

We acknowledge the limitations of our proposed method and have outlined several directions for future research. A notable issue is the randomness that arises each time NMF is performed due to the randomly initialized basis and loading factors, as well as the challenge of rank selection, which are important parameters that impact the quality of the results. As any NMF based application, the user needs to test a range of ranks and decide to choose a optimal or sub-optimal number of ranks based on the significance of the biological downstream analysis. This also applies for selecting the number of trees in the isolation forest analysis step. However, for a dataset with sufficiently large number of samples (i.e., ≥ 100), a rank of 180 and 20 trees for NMF and isolation forest was shown to be sufficient for robust results as shown in Supplementary Figure S6. For future research, DenoiseIt may be extend to true scRNA-seq samples, instead of the pseudobulk samples, or even to the recently increasing spatial scRNA-seq samples to improve cell typing and any related downstream analysis. Especially, since each cell in the single cell data is a sample, and there are thousands of cells for each cell type, DenoiseIt may be effective to capture outlier cells and remove those entirely instead of genes. Overall, we observed and conclude that gene filtering is effective in improving the robustness and quality of the downstream DEG and pathway enrichment analysis of both bulk and single-cell transcriptome samples.

Conclusion

In conclusion, DenoiseIt is an adaptable tool for the denoising gene expression data, effectively reducing noisy genes while the preserving biologically informative genes as much as possible. Its performance on four different cancer types from TCGA and a large COVID-19 cohort dataset demonstrates its capacity to retain phenotypical context associated genes that are vital for understanding specific cell types and disease conditions. Furthermore, the iterative gene set refinement procedure of DenoiseIt makes it a robust tool, suitable for a more qualitative downstream analysis. Looking ahead, DenoiseIt holds the potential to be extended beyond gene filtering to detect and exclude outlier samples, and possibly applied to scRNA-seq based spatial transcriptomics. As such, DenoiseIt is expected to aid in a more robust analysis of cohort based complex biological experiments and also serve as a tool to acquire a reliable gene set prior to any related downstream analysis.

Abbreviations

DEG	Differentially Expressed Genes
PCA	Principal Component Analysis
SVD	Singular Value Decomposition
NMF	Non-negative matrix factorization
TCGA	The Cancer Genome Atlas
COAD	Colon adenocarcinoma
STAD	Stomach adenocarcinoma
BRCA	Breast invasive carcinoma
LUAD	Lung adenocarcinoma
PBMCs	Peripheral blood mononuclear cells
NK cells	Natural Killer
WHO	World Health Organization score

TMM	Trimmed mean of M-values
NMI	Normalized Mutual Information
ARI	Adjusted Rand Index
FDR	False Discovery Rate

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-024-05899-z.

Supplementary Material 1.

Acknowledgements
This research was supported by the "Korea National Institute of Health" (KNIH) research project (project no. 2024-ER-0801-00) and the development of heterogeneous healthcare data and artificial intelligence programme (project No. 2024-NI-009-00). The authors of this paper thanks the Division of Healthcare and Artificial Intelligence group within the Korea National Institute of Health for constructing the valuable multi-omics COVID-19 cohort dataset.

Author contributions
I.J and J.J conceived the experiments, J.J and S.Y conducted the experiments, J.J, S.Y, S.K and H.J analysed the results. I.J and K.K supervised the research. All authors reviewed the manuscript.

Funding
This research was funded by the "Korea National Institute of Health" (KNIH) research project (project no. 2024-ER-0801-00) and the development of heterogeneous healthcare data and artificial intelligence programme (project No. 2024-NI-009-00). Also, this research was funded by the Infectious Disease Medical Safety, funded by the Ministry of Health and Welfare, South Korea (grant number: RS-2022-KH124555 (HG22C0014)).

Availability of data and materials
The TCGA datasets used in this study are available at the TCGA portal https://portal.gdc.cancer.gov. The datasets of the COVID-19 cohort used in this are available online in the Clinical and Omics Data Archive (CODA) database at https://coda.nih.go.kr by the accession number CODA_D23017. The DenoiseIt Software is publicly available on GitHub under the MIT license and can be found at https://github.com/cobi-git/DenoiseIt.

Declarations

Ethics approval and consent to participate
The clinical research of the COVID-19 cohort was approved by the institutional review boards of Chungnam National University Hospital (IRB no.: CNUH 2020-12-002-008), Seoul Medical Center (IRB no.: SEOUL 2021-02-016), Samsung Medical Center (IRB no.: SMC-2021-03-160), Seoul National University Hospital (IRB no.: C- 1509-103-705), and the Korea National Institute of Health (IRB no.: 2020-09-03-C-A).

Consent for publication
Not applicable

Competing interests
The authors declare no Conflict of interest.

Received: 28 March 2024 Accepted: 13 August 2024
Published online: 21 August 2024

References

1. Mohapatra P, Chakravarty S, Dash P. Microarray medical data classification using kernel ridge regression and modified cat swarm optimization based gene selection system. Swarm Evol Comput. 2016;28:144–60.
2. Liao B, Jiang Y, Liang W, Zhu W, Cai L, Cao Z. Gene selection using locality sensitive Laplacian score. IEEE/ACM Trans Comput Biol Bioinf. 2014;11(6):1146–56.
3. Wang L, Wang Y, Chang Q. Feature selection methods for big data bioinformatics: a survey from the search perspective. Methods. 2016;111:21–31.
4. Abinash, M., Vasudevan, V.: A study on wrapper-based feature selection algorithm for leukemia dataset. In: Intelligent Engineering Informatics: Proceedings of the 6th International Conference on FICTA, pp. 311–321 (2018). Springer
5. Moutsopoulos I, Maischak L, Lauzikaite E, Vasquez Urbina SA, Williams EC, Drost H-G, Mohorianu Il. noisyr: enhancing biological signal in sequencing datasets by characterizing random technical noise. Nucleic Acids Res. 2021;49(14):83–83.
6. Sha, Y., Phan, J.H., Wang, M.D.: Effect of low-expression gene filtering on detection of differentially expressed genes in rna-seq data. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 6461–6464 (2015). IEEE
7. Tabakhi S, Najafi A, Ranjbar R, Moradi P. Gene selection for microarray data classification using a novel ant colony optimization. Neurocomputing. 2015;168:1024–36.

8. Salkovic E, Sadeghi MA, Baggag A, Salem AGR, Bensmail H. Outsingle: a novel method of detecting and injecting outliers in RNA-SEQ count data using the optimal hard threshold for singular values. Bioinformatics. 2023;39(4):142.
9. Taguchi Y. Principal components analysis based unsupervised feature extraction applied to gene expression analysis of blood from dengue haemorrhagic fever patients. Sci Rep. 2017;7(1):44016.
10. Rahmanian M, Mansoori EG. An unsupervised gene selection method based on multivariate normalized mutual information of genes. Chemom Intell Lab Syst. 2022;222: 104512.
11. Liu, F.T., Ting, K.M., Zhou, Z.-H.: Isolation forest. In: 2008 Eighth IEEE international conference on data mining, pp 413–422 (2008). IEEE
12. Sanz H, Valim C, Vegas E, Oller JM, Reverter F. SVM-RFE: selection and visualization of the most relevant features through non-linear kernels. BMC Bioinform. 2018;19:1–18.
13. Filippone, M., Masulli, F., Rovetta, S.: Unsupervised gene selection and clustering using simulated annealing. In: International workshop on fuzzy logic and applications, pp 229–235 (2005). Springer
14. Tomczak K, Czerwińska P, Wiznerowicz M. Review the cancer genome atlas (TCGA): an immeasurable source of knowledge. Contemp Oncol/Współczesna Onkologia. 2015;2015(1):68–77.
15. Jo H-Y, Kim SC, Ahn D-H, Lee S, Chang S-H, Jung S-Y, Kim Y-J, Kim E, Kim J-E, Kim Y-S, et al. Establishment of the large-scale longitudinal multi-omics dataset in Covid-19 patients: data profile and biospecimen. BMB Rep. 2022;55(9):465.
16. Organization, W.H., et al.: Coronavirus disease 2019 (Covid-19): situation report, 73 (2020)
17. Hao Y, Hao S, Andersen-Nissen E, Mauck WM, Zheng S, Butler A, Lee MJ, Wilk AJ, Darby C, Zager M, et al. Integrated analysis of multimodal single-cell data. Cell. 2021;184(13):3573–87.
18. Lee DD, Seung HS. Learning the parts of objects by non-negative matrix factorization. Nature. 1999;401(6755):788–91.
19. Rand WM. Objective criteria for the evaluation of clustering methods. J Am Stat Assoc. 1971;66(336):846–50.
20. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math. 1987;20:53–65.
21. Stuart T, Butler A, Hoffman P, Hafemeister C, Papalexi E, Mauck WM, Hao Y, Stoeckius M, Smibert P, Satija R. Comprehensive integration of single-cell data. Cell. 2019;177(7):1888–902.
22. Fabregat A, Jupe S, Matthews L, Sidiropoulos K, Gillespie M, Garapati P, Haw R, Jassal B, Korninger F, May B, et al. The reactome pathway knowledgebase. Nucleic Acids Res. 2018;46(D1):649–55.
23. Gene Ontology Consortium. The gene ontology (go) database and informatics resource. Nucleic Acids Res. 2004;32(suppl_1):258–61.
24. Godec J, Tan Y, Liberzon A, Tamayo P, Bhattacharya S, Butte AJ, Mesirov JP, Haining WN. Compendium of immune signatures identifies conserved and species-specific biology in response to inflammation. Immunity. 2016;44(1):194–206.
25. Gan Y, Ye F, He X-X. The role of YWHAZ in cancer: A maze of opportunities and challenges. J Cancer. 2020;11(8):2252.
26. Naik A, Decock J. Targeting of lactate dehydrogenase c dysregulates the cell cycle and sensitizes breast cancer cells to DNA damage response targeted therapy. Mol Oncol. 2022;16(4):885–903.
27. Ciomborowska-Basheer J, Staszak K, Kubiak MR, Makałowska I. Not so dead genes-retrocopies as regulators of their disease-related progenitors and hosts. Cells. 2021;10(4):912.
28. Huang J-F, Wen C-J, Zhao G-Z, Dai Y, Li Y, Wu L-X, Zhou H-H. Overexpression of abcb4 contributes to acquired doxorubicin resistance in breast cancer cells in vitro. Cancer Chemother Pharmacol. 2018;82:199–210.
29. Chu J, Li Y, He M, Zhang H, Yang L, Yang M, Liu J, Cui C, Hong L, Hu X, et al. Zinc finger and SCAN domain containing 1, ZSCAN1, is a novel stemness-related tumor suppressor and transcriptional repressor in breast cancer targeting TAZ. Front Oncol. 2023;13:1041688.

Publisher's Note